Zabbix Manual

Welcome to the user manual for Zabbix 2.0 software. These pages are created to help our users successfully manage their monitoring tasks with Zabbix, from the simple to the more complex.

2.0/manual.txt · Last modified: 2012/05/23 11:27 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

16. Appendixes

2.0/manual/appendix.txt · Last modified: 2011/12/27 17:57 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Frequently asked questions / Troubleshooting

Frequently asked questions or FAQ.

1. Q: Can I flush/clear the queue (as depicted in *Administration → Queue*)?
   A: No.
2. Q: How do i migrate from one database to another?
   A: Dump data only (for mysql, use flag -t or --no-create-info), create the new database using schema files from Zabbix and import the data.
3. Q: I would like to replace all spaces with underscores in my item keys because they worked in older versions but space is not a valid symbol for an item key in 1.8 (or any other reason to mass-modify item keys). How should I do it and what should i beware of?
   A: You may use a database query to replace all occurrences of spaces in item keys with underscores:
   update items set key_=replace(key_,' ','_');
   Triggers will be able to use these items without any additional modifications, but you might have to change any item references in these locations:
   * Notifications (actions)
   * Map element and link labels
   * Calculated item formulas
4. Q: My graphs have dots instead of lines or empty areas. Why so?
   A: Data is missing. This can happen for a variety of reasons - performance problems on Zabbix database, Zabbix server, network, monitored devices...
5. Q: Zabbix daemons fail to start up with a message *Listener failed with error: socket() for [[-]:10050] failed with error 22: Invalid argument.*
   A: This error arises at attempt to run Zabbix agent compiled on version 2.6.27 or above on a platform with a kernel 2.6.26 and lower. Note that static linking will not help in this case because it is the socket() system call that does not support SOCK_CLOEXEC flag on earlier kernels. ZBX-3395 [https://support.zabbix.com/browse/ZBX-3395]
6. Q: I try to set up a flexible user parameter (one that accepts parameters) with a command that uses a positional parameter like $1, but it doesn't work (uses item parameter instead). How to solve this?
   A: Use a double dollar sign like **$$1**
7. Q: All dropdowns have a scrollbar and look ugly in Opera 11. Why so?
   A: It's a known bug in Opera 11.00 and 11.01; see Zabbix issue tracker [https://support.zabbix.com/browse/ZBX-3594] for more information.
8. Q: What is the structure of IDs in the database for distributed monitoring?
   A: **NNNSSSDDDDDDDDDDD**, where NNN - nodeid (to which node the ID belongs to), SSS - source nodeid (in which node was the ID created), DDDDDDDDDDD - the ID itself.
   For instance, ID created on source node 5 for node 14 might look like 1400500000012345. Note that in this example the length of ID is less than 17 digits, because the target node number has less than three digits.
9. Q: How can I change graph background colour in a custom theme?
   A: See graphs_colours table in the database and <theming guide>.

2.0/manual/appendix/faq.txt · Last modified: 2012/04/21 18:23 by zalex_ua

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

2 Installation

2.0/manual/appendix/install.txt · Last modified: 2012/02/08 16:34 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Database creation scripts

Overview

A Zabbix database must be created during the installation of Zabbix server or proxy.

This section provides scripts for creating a Zabbix database. A separate schema script is provided for each supported database.

The order of database file import has been changed - images must be imported before **data.sql** now.

For Zabbix **proxy** database **only schema.sql** should be imported (no images.sql nor data.sql)

schema.sql, **images.sql** and **data.sql** files can be found in the Zabbix source distribution, subdirectory *database*. If Zabbix was installed from distribution packages, refer to the distribution documentation.

Scripts

MySQL (we assume that user *username* with password *password* exists and has permissions to create database objects):

```
shell> mysql -u<username> -p<password>
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> quit;
shell> mysql -u<username> -p<password> zabbix < database/mysql/schema.sql
# stop here if you are creating database for Zabbix proxy
shell> mysql -u<username> -p<password> zabbix < database/mysql/images.sql
shell> mysql -u<username> -p<password> zabbix < database/mysql/data.sql
```

PostgreSQL (we assume that user *username* exists and has permissions to create database objects):

```
shell> psql -U <username>
psql> create database zabbix;
psql> \q
shell> cd database/postgresql
shell> psql -U <username> zabbix < schema.sql
# stop here if you are creating database for Zabbix proxy
shell> psql -U <username> zabbix < images.sql
shell> psql -U <username> zabbix < data.sql
```

Oracle (we assume that user *zabbix* with password *password* exists and has permissions to create database objects in service ORCL):

If you are creating database for Zabbix server you need to have images somewhere on Oracle host:

```
# copy directory create/images/old_images somewhere on Oracle server (e.g. /home/oracle):
shell> scp -r create/images/old_images user@host:/home/oracle
# edit file database/oracle/images.sql and set images_dir variable to where you have copied them (e. g. "/home/oracle/old_images"):
CREATE OR REPLACE DIRECTORY image_dir AS '/home/oracle/old_images'
```

Now prepare the database:

```
shell> sqlplus zabbix/password@host/ORCL
sqlplus> set def off
sqlplus> @database/oracle/schema.sql
# stop here if you are creating database for Zabbix proxy
sqlplus> @database/oracle/images.sql
sqlplus> @database/oracle/data.sql
```

Zabbix requires a UTF-8 database character set. If database is not UTF-8, it can be converted by running: `ALTER DATABASE NATIONAL CHARACTER SET UTF8;`

IBM DB2:

```
shell> db2 "create database zabbix using codeset utf-8 territory us pagesize 32768"
shell> cd database/ibm_db2
shell> db2batch -d zabbix -f schema.sql
# stop here if you are creating database for Zabbix proxy
shell> db2batch -d zabbix -f images.sql
shell> db2batch -d zabbix -f data.sql
```

SQLite:

```
shell> cd database/sqlite3
shell> sqlite3 /var/lib/sqlite/zabbix.db < schema.sql
# stop here if you are creating database for Zabbix proxy
shell> sqlite3 /var/lib/sqlite/zabbix.db < images.sql
shell> sqlite3 /var/lib/sqlite/zabbix.db < data.sql
```

If using SQLite with Zabbix proxy, database will be automatically created if it does not exist.

Return to installation section.

2.0/manual/appendix/install/db_scripts.txt · Last modified: 2013/06/04 10:03 by jaba

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

2 Zabbix Windows agent installation

Installation

Installation is very simple and includes 3 steps.

Step 1

Create a configuration file.

Create a configuration file c:/zabbix_agentd.conf (it has a similar syntax to the UNIX agent configuration file).

An example configuration file is available in Zabbix source archive as *conf/zabbix_agentd.win.conf*.

Step 2

Install agent as a Windows service.

```
zabbix_agentd.exe --install
```

On a 64-bit system, a 64-bit Zabbix agent version is required for all checks related to running 64-bit processes to work correctly.

If you wish to use a configuration file other than c:\zabbix_agentd.conf, you should use the following command for service installation:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

A full path to the configuration file should be specified.

Step 3

Run agent.

Now you can use Control Panel to start the agent service or run:

```
zabbix_agentd.exe --start
```

Configuration

Hostname parameter

To perform active checks on a host Zabbix agent needs to have the hostname defined. Moreover, the hostname value set on the agent side should exactly match the "Host name" configured for the host in the frontend.

The hostname value on the agent side can be defined by either the *Hostname* or *HostnameItem* parameter in the agent configuration file - or the default values are used if any of these parameters are not specified.

The default value for *HostnameItem* parameter is the value returned by the "system.hostname" agent key and for Windows platform it returns the NetBIOS host name.

The default value for *Hostname* is the value returned by the *HostnameItem* parameter. So, in effect, if both these parameters are unspecified the actual hostname will be the host NetBIOS name.

The **system.hostname** key always returns the NetBIOS host name which is limited to 15 symbols and in UPPERCASE only - regardless of the length and lowercase/uppercase characters in the real host name.

So if both parameters are unspecified then the Zabbix agent will use NetBIOS host name to retrieve the list of active checks from the Zabbix server and to send results to it.

Starting from Zabbix agent 1.8.6 version for Windows the “system.hostname” key supports an optional parameter - *type* of the name. The default value of this parameter is “netbios” (for backward compatibility) and the other possible value is “host”.

The **system.hostname[host]** key always returns the full, real (case sensitive) Windows host name.

So, to simplify the configuration of zabbix_agentd.conf file and make it unified, two different approaches could be used.

1. leave *Hostname* or *HostnameItem* parameters undefined and Zabbix agent will use NetBIOS host name as the hostname;
2. leave *Hostname* parameter undefined and define *HostnameItem* like this:
   HostnameItem=system.hostname[host]
   and the Zabbix agent will use the full, real (case sensitive) Windows host name as the hostname.

2.0/manual/appendix/install/windows_agent.txt · Last modified: 2013/04/25 11:41 by martins-v
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

3 Daemon configuration

2.0/manual/appendix/config.txt · Last modified: 2012/02/20 15:13 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Zabbix server

The parameters supported in a Zabbix server configuration file:

Parameter	Mandatory	Range	Default	Description
AlertScriptsPath	no		/usr/local/share/zabbix/alertscripts	Location of custom alert scripts (depends on compile-time installation variable *datadir*).
CacheSize	no	128K-2G	8M	Size of configuration cache, in bytes. Shared memory size for storing host, item and trigger data. The upper limit used to be 1G before version 2.0.5.
CacheUpdateFrequency	no	1-3600	60	How often Zabbix will perform update of configuration cache, in seconds.
DBHost	no		localhost	Database host name. In case of MySQL localhost or empty string results in using a socket. In case of PostgreSQL only empty string results in attempt to use socket.
DBName	yes			Database name. For SQLite3 path to database file must be provided. DBUser and DBPassword are ignored.
DBPassword	no			Database password. Ignored for SQLite. Comment this line if no password is used.
DBPort	no	1024-65535	3306	Database port when not using local socket. Ignored for SQLite.
DBSchema	no			Schema name. Used for IBM DB2.
DBSocket	no		/tmp/mysql.sock	Path to MySQL socket.
DBUser	no			Database user. Ignored for SQLite.
DebugLevel	no	0-4	3	Specifies debug level: 0 - no debug 1 - critical information 2 - error information 3 - warnings 4 - for debugging (produces lots of information)
DisableHousekeeping	no	0-1	0	If set to 1, housekeeping is disabled.
ExternalScripts	no		/usr/local/share/zabbix/externalscripts	Location of external scripts (depends on compile-time installation variable *datadir*).
Fping6Location	no		/usr/sbin/fping6	Location of fping6. Make sure that fping6 binary has root ownership and SUID flag set. Make empty (“Fping6Location=”) if your fping utility is capable to process IPv6 addresses.
FpingLocation	no		/usr/sbin/fping	Location of fping. Make sure that fping binary has root ownership and SUID flag set!

HistoryCacheSize	no	128K-2G	8M	Size of history cache, in bytes. Shared memory size for storing history data. The upper limit used to be 1G before version 2.0.5.
HistoryTextCacheSize	no	128K-2G	16M	Size of text history cache, in bytes. Shared memory size for storing character, text or log history data. The upper limit used to be 1G before version 2.0.5.
HousekeepingFrequency	no	1-24	1	How often Zabbix will perform housekeeping procedure (in hours). Housekeeping is removing unnecessary information from history, alert, and alarms tables.
MaxHousekeeperDelete	no	0-1000000	500	No more than 'MaxHousekeeperDelete' rows (corresponding to [tablename], [field], [value]) will be deleted per one task in one housekeeping cycle. SQLite3 does not use this parameter, deletes all corresponding rows without a limit. If set to 0 then no limit is used at all. In this case you must know what you are doing! **This option is supported in version 1.8.2 and higher.**
Include	no			You may include individual files or all files in a directory in the configuration file.
JavaGateway	no			IP address (or hostname) of Zabbix Java gateway. Only required if Java pollers are started. **This option is supported in version 2.0.0 and higher.**
JavaGatewayPort	no	1024-32767	10052	Port that Zabbix Java gateway listens on. **This option is supported in version 2.0.0 and higher.**
ListenIP	no		0.0.0.0	List of comma delimited IP addresses that the trapper should listen on. Trapper will listen on all network interfaces if this parameter is missing. **Multiple IP addresses are supported in version 1.8.3 and higher.**
ListenPort	no	1024-32767	10051	Listen port for trapper.
LogFileSize	no	0-1024	1	Maximum size of log file in MB. 0 - disable automatic log rotation.
LogFile	no			Name of log file.
LogSlowQueries	no	0-3600000	0	How long a database query may take before being logged (in milliseconds). 0 - don't log slow queries. This option becomes enabled starting with DebugLevel=3. **Supported since Zabbix 1.8.2 version.**
				Unique NodeID in distributed setup.

NodeID	no	0-999	0	0 - standalone server
NodeNoEvents	no	0-1	0	If set to '1' local events won't be sent to master node. This won't impact ability of this node to propagate events from its child nodes.
NodeNoHistory	no	0-1	0	If set to '1' local history won't be sent to master node. This won't impact ability of this node to propagate history from its child nodes.
PidFile	no		/tmp/zabbix_server.pid	Name of PID file.
ProxyConfigFrequency	no	1-604800	3600	How often Zabbix server sends configuration data to a Zabbix proxy in seconds. Used only for proxies in a passive mode. **This option is supported in version 1.8.3 and higher.**
ProxyDataFrequency	no	1-3600	1	How often Zabbix server requests history data from a Zabbix proxy in seconds. Used only for proxies in a passive mode. **This option is supported in version 1.8.3 and higher.**
SenderFrequency	no	5-3600	30	How often Zabbix will try to send unsent alerts (in seconds).
SNMPTrapperFile	no		/tmp/zabbix_traps.tmp	Temporary file used for passing data from SNMP trap daemon to the server. Must be the same as in zabbix_trap_receiver.pl or SNMPTT configuration file. **This option is supported in version 2.0.0 and higher.**
SourceIP	no			Source IP address for outgoing connections.
SSHKeyLocation	no			Location of public and private keys for SSH checks and actions
StartDBSyncers	no	1-100	4	Number of pre-forked instances of DB Syncers. The upper limit used to be 64 before version 1.8.5. **This option is supported in version 1.8.3 and higher.**
StartDiscoverers	no	0-250	1	Number of pre-forked instances of discoverers. The upper limit used to be 255 before version 1.8.5.
StartHTTPPollers	no	0-1000	1	Number of pre-forked instances of HTTP pollers. The upper limit used to be 255 before version 1.8.5.
StartIPMIPollers	no	0-1000	0	Number of pre-forked instances of IPMI pollers. The upper limit used to be 255 before version 1.8.5.
				Number of pre-forked instances of Java pollers.

StartJavaPollers	no	0-1000	0	**This option is supported in version 2.0.0 and higher.**
StartPingers	no	0-1000	1	Number of pre-forked instances of ICMP pingers. The upper limit used to be 255 before version 1.8.5.
StartPollersUnreachable	no	0-1000	1	Number of pre-forked instances of pollers for unreachable hosts (including IPMI). The upper limit used to be 255 before version 1.8.5. **This option is missing in version 1.8.3.**
StartPollers	no	0-1000	5	Number of pre-forked instances of pollers. The upper limit used to be 255 before version 1.8.5.
StartProxyPollers	no	0-250	1	Number of pre-forked instances of pollers for passive proxies. The upper limit used to be 255 before version 1.8.5. **This option is supported in version 1.8.3 and higher.**
StartSNMPTrapper	no	0-1	0	If set to 1, SNMP trapper process will be started. **This option is supported in version 2.0.0 and higher.**
StartTrappers	no	0-1000	5	Number of pre-forked instances of trappers. Trappers accept incoming connections from Zabbix sender, active agents, active proxies and child nodes. At least one trapper process must be running to display server availability in the frontend. The upper limit used to be 255 before version 1.8.5.
Timeout	no	1-30	3	Specifies how long we wait for agent, SNMP device or external check (in seconds).
TmpDir	no		/tmp	Temporary directory.
TrapperTimeout	no	1-300	300	Specifies how many seconds trapper may spend processing new data.
TrendCacheSize	no	128K-2G	4M	Size of trend cache, in bytes. Shared memory size for storing trends data. The upper limit used to be 1G before version 2.0.5.
UnavailableDelay	no	1-3600	60	How often host is checked for availability during the unavailability period, in seconds.
UnreachableDelay	no	1-3600	15	How often host is checked for availability during the unreachability period, in seconds.
UnreachablePeriod	no	1-3600	45	After how many seconds of unreachability treat a host as unavailable.

2.0/manual/appendix/config/zabbix_server.txt · Last modified: 2013/04/30 10:30 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

2 Zabbix proxy

The parameters supported in a Zabbix proxy configuration file:

Parameter	Mandatory	Range	Default	Description
CacheSize	no	128K-2G	8M	Size of configuration cache, in bytes. Shared memory size, for storing hosts and items data. The upper limit used to be 1G before version 2.0.5.
ConfigFrequency	no	1-604800	3600	How often proxy retrieves configuration data from Zabbix server in seconds. For a proxy in the passive mode this parameter will be ignored.
DBHost	no		localhost	Database host name. In case of MySQL localhost or empty string results in using a socket. In case of PostgreSQL only empty string results in attempt to use socket.
DBName	yes			Database name. For SQLite3 path to database file must be provided. DBUser and DBPassword are ignored. Warning: Do not attempt to use the same database Zabbix server is using.
DBPassword	no			Database password. Ignored for SQLite. Comment this line if no password is used.
DBSchema	no			Schema name. Used for IBM DB2.
DBSocket	no		3306	Path to MySQL socket. Database port when not using local socket. Ignored for SQLite.
DBUser				Database user. Ignored for SQLite.
DataSenderFrequency	no	1-3600	1	Proxy will send collected data to the server every N seconds.
DebugLevel	no	0-4	3	Specifies debug level: 0 - no debug 1 - critical information 2 - error information 3 - warnings 4 - for debugging (produces lots of information)
ExternalScripts	no		/usr/local/share/zabbix/externalscripts	Location of external scripts (depends on compile-time installation variable *datadir*).
Fping6Location	no		/usr/sbin/fping6	Location of fping6. Make sure that fping6 binary has root ownership and SUID flag set. Make empty (“Fping6Location=”) if your fping utility is capable to process IPv6 addresses.
				Location of fping.

FpingLocation	no		/usr/sbin/fping	Make sure that fping binary has root ownership and SUID flag set!
HeartbeatFrequency	no	0-3600	60	Frequency of heartbeat messages in seconds. Used for monitoring availability of proxy on server side. 0 - heartbeat messages disabled. For a proxy in the passive mode this parameter will be ignored.
HistoryCacheSize	no	128K-2G	8M	Size of history cache, in bytes. Shared memory size for storing history data. The upper limit used to be 1G before version 2.0.5.
HistoryTextCacheSize	no	128K-2G	16M	Size of text history cache, in bytes. Shared memory size for storing character, text or log history data. The upper limit used to be 1G before version 2.0.5.
Hostname	no		Set by HostnameItem	Unique, case sensitive Proxy name. Make sure the proxy name is known to the server! Allowed characters: alphanumeric, '.', ' ', '_' and '-'. Maximum length: 64
HostnameItem	no		system.hostname	Item used for setting Hostname if it is undefined (this will be run on the proxy similarly as on an agent). Does not support UserParameters, performance counters or aliases, but does support system.run[]. Ignored if Hostname is set. **This option is supported in version 1.8.6 and higher.**
HousekeepingFrequency	no	1-24	1	How often Zabbix will perform housekeeping procedure (in hours). Housekeeping is removing unnecessary information from history, alert, and alarms tables.
Include	no			You may include individual files or all files in a directory in the configuration file.
JavaGateway	no			IP address (or hostname) of Zabbix Java gateway. Only required if Java pollers are started. **This option is supported in version 2.0.0 and higher.**
JavaGatewayPort	no	1024-32767	10052	Port that Zabbix Java gateway listens on. **This option is supported in version 2.0.0 and higher.**
ListenIP	no		0.0.0.0	List of comma delimited IP addresses that the trapper should listen on. Trapper will listen on all network interfaces if this parameter is missing. **Multiple IP addresses are supported in version 1.8.3 and higher.**

ListenPort	no	1024-32767	10051	Listen port for trapper.
LogFileSize	no	0-1024	1	Maximum size of log file in MB. 0 - disable automatic log rotation.
LogFile	no			Name of log file. If not set, syslog is used.
LogSlowQueries	no	0-3600000	0	How long a database query may take before being logged (in milliseconds). 0 - don't log slow queries. This option becomes enabled starting with DebugLevel=3. **Supported since Zabbix 1.8.2 version.**
PidFile	no		/tmp/zabbix_proxy.pid	Name of PID file.
ProxyLocalBuffer	no	0-720	0	Proxy will keep data locally for N hours, even if the data have already been synced with the server. This parameter may be used if local data will be used by third party applications.
ProxyMode	no	0-1	0	Proxy operating mode. 0 - proxy in the active mode 1 - proxy in the passive mode **This option is supported in version 1.8.3 and higher.**
ProxyOfflineBuffer	no	1-720	1	Proxy will keep data for N hours in case if no connectivity with Zabbix server. Older data will be lost.
ServerPort	no	1024-32767	10051	Port of Zabbix trapper on Zabbix server. For a proxy in the passive mode this parameter will be ignored.
Server	yes			IP address (or hostname) of Zabbix server. Active proxy will get configuration data from the server. For a proxy in the passive mode this parameter will be ignored.
SNMPTrapperFile	no		/tmp/zabbix_traps.tmp	Temporary file used for passing data from SNMP trap daemon to the proxy. Must be the same as in zabbix_trap_receiver.pl or SNMPTT configuration file. **This option is supported in version 2.0.0 and higher.**
SourceIP	no			Source IP address for outgoing connections.
SSHKeyLocation	no			Location of public and private keys for SSH checks and actions
StartDBSyncers	no	1-100	4	Number of pre-forked instances of DB Syncers. The upper limit used to be 64 before version 1.8.5. **This option is supported in version 1.8.3 and higher.**
StartDiscoverers	no	0-250	1	Number of pre-forked instances of discoverers.

				The upper limit used to be 255 before version 1.8.5.
StartHTTPPollers	no	0-1000	1	Number of pre-forked instances of HTTP pollers.
StartIPMIPollers	no	0-1000	0	Number of pre-forked instances of IPMI pollers. The upper limit used to be 255 before version 1.8.5.
StartJavaPollers	no	0-1000	0	Number of pre-forked instances of Java pollers. **This option is supported in version 2.0.0 and higher.**
StartPingers	no	0-1000	1	Number of pre-forked instances of ICMP pingers. The upper limit used to be 255 before version 1.8.5.
StartPollersUnreachable	no	0-1000	1	Number of pre-forked instances of pollers for unreachable hosts (including IPMI). The upper limit used to be 255 before version 1.8.5. **This option is missing in version 1.8.3.**
StartPollers	no	0-1000	5	Number of pre-forked instances of pollers. The upper limit used to be 255 before version 1.8.5.
StartSNMPTrapper	no	0-1	0	If set to 1, SNMP trapper process will be started. **This option is supported in version 2.0.0 and higher.**
StartTrappers	no	0-1000	5	Number of pre-forked instances of trappers. Trappers accept incoming connections from Zabbix sender and active agents. The upper limit used to be 255 before version 1.8.5.
Timeout	no	1-30	3	Specifies how long we wait for agent, SNMP device or external check (in seconds).
TmpDir	no		/tmp	Temporary directory.
TrapperTimeout	no	1-300	300	Specifies how many seconds trapper may spend processing new data.
UnavailableDelay	no	1-3600	60	How often host is checked for availability during the unavailability period, in seconds.
UnreachableDelay	no	1-3600	15	How often host is checked for availability during the unreachability period, in seconds.
UnreachablePeriod	no	1-3600	45	After how many seconds of unreachability treat a host as unavailable.

2.0/manual/appendix/config/zabbix_proxy.txt · Last modified: 2013/10/09 22:28 by richlv

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

3 Zabbix agent (UNIX)

The parameters supported in a Zabbix agent configuration file (*zabbix_agentd.conf*):

Parameter	Mandatory	Range	Default	Description
Alias	no			Sets an alias for an item key. The alias can be used instead of the full item key in item configuration. For example: *Alias*=system.cpu.load[percpu,5]:system.cpu.load[percpu,avg5] *Item key*: system.cpu.load[percpu,5] *Item name*: Processor load ($2 min average per core) An alias can be especially useful to substitute a long and complex item key with a smaller and simpler one. Multiple aliases can be defined.
AllowRoot	no		0	Allow the agent to run as 'root'. If disabled and the agent is started by 'root', the agent will try to switch to user 'zabbix' instead. Has no effect if started under a regular user. 0 - do not allow 1 - allow
BufferSend	no	1-3600	5	Do not keep data longer than N seconds in buffer.
BufferSize	no	2-65535	100	Maximum number of values in a memory buffer. The agent will send all collected data to Zabbix server or proxy if the buffer is full.
DebugLevel	no	0-4	3	Specifies debug level 0 - no debug 1 - critical information 2 - error information 3 - warnings 4 - for debugging (produces lots of information)
EnableRemoteCommands	no		0	Whether remote commands from Zabbix server are allowed. 0 - not allowed 1 - allowed
Hostname	no		Set by HostnameItem	Unique, case sensitive hostname. Required for active checks and must match hostname as configured on the server. Allowed characters: alphanumeric, '.', ' ', '_' and '-'. Maximum length: 64
HostnameItem	no		system.hostname	Item used for setting Hostname if it is undefined. Does not support UserParameters, performance counters or aliases, but does support system.run[] regardless of EnableRemoteCommands value. Ignored if Hostname is set. **This option is supported in version 1.8.6 and higher.**
Include	no			You may include individual files or all files in a directory in the configuration file.
ListenIP	no		0.0.0.0	List of comma delimited IP addresses that the agent should listen on. **Multiple IP addresses are supported in version 1.8.3 and higher.**
ListenPort	no	1024-32767	10050	Agent will listen on this port for connections from the server.
LogFileSize	no	0-1024	1	Maximum size of log file in MB. 0 - disable automatic log rotation.
LogFile	no			Name of log file. If not set, syslog is used.

LogRemoteCommands	no		0	Enable logging of executed shell commands as warnings. 0 - disabled 1 - enabled
MaxLinesPerSecond	no	1-1000	100	Maximum number of new lines the agent will send per second to Zabbix server or proxy when processing 'log' and 'eventlog' active checks. The provided value will be overridden by the parameter 'maxlines', provided in 'log' or 'eventlog' item key. *Note*: Zabbix will process 4 times more new lines than set in *MaxLinesPerSecond* to seek the required string in log items.
PidFile	no		/tmp/zabbix_agentd.pid	Name of PID file.
RefreshActiveChecks	no	60-3600	120	How often list of active checks is refreshed, in seconds. Note that after failing to refresh active checks the next refresh will be attempted after 60 seconds.
Server	no			List of comma delimited IP addresses (or hostnames) of Zabbix servers. No spaces allowed. Incoming connections will be accepted only from the hosts listed here. Note that hostnames must resolve hostname→IP address and IP address→hostname. If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally.
ServerActive	no			List of comma delimited IP:port (or hostname:port) pairs of Zabbix servers for active checks. No spaces allowed. If port is not specified, default port is used. IPv6 addresses must be enclosed in square brackets if port for that host is specified. If port is not specified, square brackets for IPv6 addresses are optional. If this parameter is not specified, active checks are disabled.
SourceIP	no			Source IP address for outgoing connections.
StartAgents	no	0-100	3	Number of pre-forked instances of zabbix_agentd that process passive checks. If set to 0, disables passive checks and the agent will not listen on any TCP port. The upper limit used to be 16 before version 1.8.5.
Timeout	no	1-30	3	Spend no more than Timeout seconds on processing
UnsafeUserParameters	no	0,1	0	Allow all characters to be passed in arguments to user-defined parameters. Supported since Zabbix 1.8.2.
UserParameter	no			User-defined parameter to monitor. There can be several user-defined parameters. Format: UserParameter=<key>,<shell command> Note that shell command must not return empty string or EOL only. Example: UserParameter=system.test,who|wc -l

In Zabbix agent 2.0.0 version configuration parameters related to active and passive checks have been changed. See the "See also" section at the bottom of this page to read more details about these changes.

See also

1. Differences in the Zabbix agent configuration for active and passive checks starting from version 2.0.0 [http://blog.zabbix.com/multiple-servers-for-active-agent-sure/858]

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-

Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

4 Zabbix agent (Windows)

The parameters supported in a Zabbix agent (Windows) configuration file:

Parameter	Mandatory	Range	Default	Description
Alias	no			Sets an alias for an item key. The alias can be used instead of the full item key in item configuration. For example, to retrieve paging file usage percentage from the server, you may use the following item key: perf_counter[\Paging File(_Total)\% Usage] However, you may also define an alias by adding the following line to configuration file: *Alias*=pg_usage:perf_counter[\Paging File(_Total)\% Usage] and then use “pg_usage” instead of the full item key in item configuration. An alias can be especially useful to substitute a long and complex item key with a smaller and simpler one. Multiple aliases can be defined. Aliases cannot be reused in PerfCounter parameters.
BufferSend	no	1-3600	5	Do not keep data longer than N seconds in buffer.
BufferSize	no	2-65535	100	Maximum number of values in a memory buffer. The agent will send all collected data to Zabbix server or proxy if the buffer is full.
DebugLevel	no	0-4	3	Specifies debug level 0 - no debug 1 - critical information 2 - error information 3 - warnings 4 - for debugging (produces lots of information)
EnableRemoteCommands	no		0	Whether remote commands from Zabbix server are allowed. 0 - not allowed 1 - allowed
Hostname	no		Set by HostnameItem	Unique, case sensitive hostname. Required for active checks and must match hostname as configured on the server. Allowed characters: alphanumeric, '.', ' ', '_' and '-'. Maximum length: 64
HostnameItem	no		system.hostname	Item used for setting Hostname if it is undefined. Does not support UserParameters, performance counters or aliases, but does support system.run[] regardless of EnableRemoteCommands value. Ignored if Hostname is set. **This option is supported in version 1.8.6 and higher.** See also a more detailed description.
Include	no			You may include individual file in the configuration file.
ListenIP	no		0.0.0.0	Agent will listen on the specified interface.
ListenPort	no	1024-32767	10050	Agent will listen on this port for connections from the server.
LogFileSize	no	0-1024	1	Maximum size of log file in MB. 0 - disable automatic log rotation.
LogFile	no			Name of log file.

				If not set, Windows Event Log is used.
LogRemoteCommands	no		0	Enable logging of executed shell commands as warnings. 0 - disabled 1 - enabled
MaxLinesPerSecond	no	1-1000	100	Maximum number of new lines the agent will send per second to Zabbix server or proxy processing 'log', 'logrt' and 'eventlog' active checks. The provided value will be overridden by the parameter 'maxlines', provided in 'log', 'logrt' or 'eventlog' item keys.
PerfCounter	no			Syntax: <parameter_name>,"<perf_counter_path>",<period> Defines new parameter <parameter_name> which is an average value for system performance counter <perf_counter_path> for the specified time period <period> (in seconds). For example, if you wish to receive average number of processor interrupts per second for last minute, you can define new parameter "interrupts" as following: PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60 Please note double quotes around performance counter path. Samples for calculating average value will be taken every second. You may run "typeperf -qx" to get list of all performance counters available in Windows.
RefreshActiveChecks	no	60-3600	120	How often list of active checks is refreshed, in seconds. Note that after failing to refresh active checks the next refresh will be attempted after 60 seconds.
Server	no			List of comma delimited IP addresses (or hostnames) of Zabbix servers. No spaces allowed. Incoming connections will be accepted only from the hosts listed here. Note that hostnames must resolve hostname→IP address and IP address→hostname. If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally.
ServerActive	no			List of comma delimited IP:port (or hostname:port) pairs of Zabbix servers for active checks. No spaces allowed. If port is not specified, default port is used. IPv6 addresses must be enclosed in square brackets if port for that host is specified. If port is not specified, square brackets for IPv6 addresses are optional. If this parameter is not specified, active checks are disabled.
SourceIP	no			Source IP address for outgoing connections.
StartAgents	no	0-100	3	Number of pre-forked instances of zabbix_agentd that process passive checks. If set to 0, disables passive checks and the agent will not listen on any TCP port. The upper limit used to be 16 before version 1.8.5.
Timeout	no	1-30	3	Spend no more than Timeout seconds on processing
UnsafeUserParameters	no	0-1	0	Allow all characters to be passed in arguments to user-defined parameters. 0 - do not allow 1 - allow
UserParameter				User-defined parameter to monitor. There can be several user-defined parameters. Format: UserParameter=<key>,<shell command> Note that shell command must not return empty string or EOL only. Example: UserParameter=system.test,echo 1

In Zabbix agent 2.0.0 version configuration parameters related to active and passive checks have been changed.
See the "See also" section at the bottom of this page to read more details about these changes.

See also

1. Differences in the Zabbix agent configuration for active and passive checks starting from version 2.0.0 [http://blog.zabbix.com/multiple-servers-for-active-agent-sure/858] configuration file records. |

2.0/manual/appendix/config/zabbix_agentd_win.txt · Last modified: 2013/09/24 09:34 by martins-v
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

5 Zabbix Java gateway

If you use `startup.sh` and `shutdown.sh` scripts for starting Zabbix Java gateway, then you can specify the necessary configuration parameters in file `settings.sh`. The startup and shutdown scripts source the settings file and take care of converting shell variables (listed in the first column) to Java properties (listed in the second column).

If you start Zabbix Java gateway manually by running `java` directly, then you specify the corresponding Java properties on the command line.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	IP address to listen on.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Port to listen on.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Name of PID file. If omitted, Zabbix Java Gateway is started as a console application.
START_POLLERS	zabbix.startPollers	no	1-1000	5	Number of worker threads to start.

Port 10052 is not IANA registered [http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt].

2.0/manual/appendix/config/zabbix_java.txt · Last modified: 2012/04/16 16:12 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

6 Archive: Zabbix agent (UNIX, Inetd version)

The parameters supported in a Zabbix agent (UNIX, Inetd version) configuration file:

Parameter	Mandatory	Default value	Description
Alias	no		Sets an alias for an item key. The alias can be used instead of the full item key in item configuration. For example: *Alias*=system.cpu.load[percpu,5]:system.cpu.load[percpu,avg5] *Item key*: system.cpu.load[percpu,5] *Item name*: Processor load ($2 min average per core) An alias can be especially useful to substitute a long and complex item key with a smaller and simpler one. Multiple aliases can be defined.
Include	no		You may include individual files or all files in a directory in the configuration file.
Server	yes	-	Comma-delimited list of IP addresses of Zabbix servers or proxies. Connections from other IP addresses will be rejected.
Timeout	no	3	Do not spend more than Timeout seconds on getting requested value (1-255). The agent does not kill timeouted user parameter processes!
UnsafeUserParameters	no	0	Allow all characters to be passed in arguments to user-defined parameters
UserParameter	no		User-defined parameter to monitor. There can be several user-defined parameters. Format: UserParameter=<key>,<shell command> Note that shell command must not return empty string or EOL only. Example: UserParameter=system.test,who|wc -l

2.0/manual/appendix/config/zabbix_agent.txt · Last modified: 2013/09/23 11:04 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

4 Items

2.0/manual/appendix/items.txt · Last modified: 2012/02/20 15:14 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Items supported by platform

The table displays support for Zabbix agent items on various platforms:

- Items marked with "**X**" are supported, the ones marked with "**-**" are not supported.
- If an item is marked with "**?**", it is not known whether it is supported or not.
- If an item is marked with "**r**", it means that it requires root privileges.
- If a parameter is marked with "**i**", it means that it is ignored.
- Parameters that are included in angle brackets <like_this> are optional.

Windows-only Zabbix agent items are not included in this table.

Parameter / system		Windows ▼▼	Linux 2.4 ▼▼	Linux 2.6 ▼▼	FreeBSD ▼▼	Solaris ▼▼	HP-UX ▼▼	AIX ▼▼	Tru64 ▼▼	Mac OS X ▼▼	OpenBSD ▼▼	NetBSD ▼▼
▼▼		1	2	3	4	5	6	7	8	9	10	11
agent.hostname		X	X	X	X	X	X	X	X	X	X	X
agent.ping		X	X	X	X	X	X	X	X	X	X	X
agent.version		X	X	X	X	X	X	X	X	X	X	X
kernel.maxfiles		-	X	X	X	-	-	-	?	X	X	X
kernel.maxproc		-	-	X	X	X	-	-	?	X	X	X
log[file,<regexp>,<encoding>,<maxlines>,<mode>]		X	X	X	X	X	X	X	X	X	X	X
logrt[file_format,<regexp>,<encoding>,<maxlines>,<mode>]		X	X	X	X	X	X	X	X	X	X	X
net.dns[<ip>,zone,<type>,<timeout>,<count>]		X	X	X	X	X	X	X	X	X	X	X
net.dns.record[<ip>,zone,<type>,<timeout>,<count>]		X	X	X	X	X	X	X	X	X	X	X
net.if.collisions[if]		-	X	X	X	X	-	X	-	X	X	r
net.if.discovery		X	X	X	-	X	X	X	-	-	-	-
net.if.in[if,<mode>]		X	X	X	X	X	-	X	-	X	X	r
mode ▲	bytes *(default)*	X	X	X	X	X	-	X	-	X	X	r
	packets	X	X	X	X	X	-	X	-	X	X	r
	errors	X	X	X	X	X	-	X	-	X	X	r
	dropped	X	X	X	X	-	-	-	-	X	X	r
net.if.out[if,<mode>]		X	X	X	X	X	-	X	-	X	X	r
	bytes *(default)*	X	X	X	X	X	-	X	-	X	X	r

mode ▲	packets	X	X	X	X	X	–	X	–	X	X	r
	errors	X	X	X	X	X	–	X	–	X	X	r
	dropped	X	X	X	–	–	–	–	–	–	–	–
net.if.total[if,<mode>]		X	X	X	X	X	–	X	–	X	X	r
mode ▲	bytes *(default)*	X	X	X	X	X	–	X	–	X	X	r
	packets	X	X	X	X	X	–	X	–	X	X	r
	errors	X	X	X	X	X	–	X	–	X	X	r
	dropped	X	X	X	–	–	–	–	–	–	–	–
net.tcp.listen[port]		X	X	X	X	X	–	–	–	X	–	–
net.tcp.port[<ip>,port]		X	X	X	X	X	X	X	X	X	X	X
net.tcp.service[service,<ip>,<port>]		X	X	X	X	X	X	X	X	X	X	X
net.tcp.service.perf[service,<ip>,<port>]		X	X	X	X	X	X	X	X	X	X	X
net.udp.listen[port]		–	X	X	X	X	–	–	–	X	–	–
		1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11
proc.mem[<name>,<user>,<mode>,<cmdline>]		–	X	X	X	X	–	X	X	–	X	X
mode ▲	sum *(default)*	–	X	X	X	X	–	X	X	–	X	X
	avg	–	X	X	X	X	–	X	X	–	X	X
	max	–	X	X	X	X	–	X	X	–	X	X
	min	–	X	X	X	X	–	X	X	–	X	X
proc.num[<name>,<user>,<state>,<cmdline>]		X	X	X	X	X	–	X	X	–	X	X
state ▲	all *(default)*	–	X	X	X	X	–	X	X	–	X	X
	sleep	–	X	X	X	X	–	X	X	–	X	X
	zomb	–	X	X	X	X	–	X	X	–	X	X
	run	–	X	X	X	X	–	X	X	–	X	X
cmdline ▲		–	X	X	X	X	–	X	X	–	X	X
sensor[device,sensor,<mode>]		–	X	–	–	–	–	–	–	–	X	–
system.boottime		–	X	X	X	X	–	–	–	X	X	X
system.cpu.intr		–	X	X	X	X	–	X	–	–	X	X
system.cpu.load[<cpu>,<mode>]		X	X	X	X	X	X	X	X	X	X	X
cpu ▲	all *(default)*	X	X	X	X	X	X	X	X	X	X	X
	percpu	X	X	X	X	X	X	X	–	X	X	X
mode ▲	avg1 *(default)*	X	X	X	X	X	X	X	X	X	X	X
	avg5	X	X	X	X	X	X	X	X	X	X	X
	avg15	X	X	X	X	X	X	X	X	X	X	X
system.cpu.num[<type>]		X	X	X	X	X	X	X	–	X	X	X
type ▲	online *(default)*	X	X	X	X	X	X	X	–	X	X	X
	max	–	X	X	X	X	–	–	–	X	–	–
system.cpu.switches		–	X	X	X	X	–	X	–	–	X	X
system.cpu.util[<cpu>,<type>,<mode>]		X	X	X	X	X	X	X	X	–	X	X
	user *(non-Windows default)*	–	X	X	X	X	X	X	X	–	X	X
	nice	–	X	X	X	–	X	–	X	–	X	X
	idle	–	X	X	X	X	X	X	X	–	X	X
	system *(Windows default)*	X	X	X	X	X	X	X	X	–	X	X

type ▲												
	iowait	–	–	X	–	X	–	X	–	–	–	–
	interrupt	–	–	X	X	–	–	–	–	–	X	–
	softirq	–	–	X	–	–	–	–	–	–	–	–
	steal	–	–	X	–	–	–	–	–	–	–	–
mode ▲	avg1 *(default)*	X	X	X	X	X	X	X	X	–	X	X
	avg5	X	X	X	X	X	X	X	–	–	X	X
	avg15	X	X	X	X	X	X	X	–	–	X	X
		1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11
system.hostname[<type>]		X	X	X	X	X	X	X	X	X	X	X
system.hw.chassis[<info>]		–	X	X	–	–	–	–	–	–	–	–
system.hw.cpu[<cpu>,<info>]		–	X	X	–	–	–	–	–	–	–	–
system.hw.devices[<type>]		–	X	X	–	–	–	–	–	–	–	–
system.hw.macaddr[<interface>,<format>]		–	X	X	–	–	–	–	–	–	–	–
system.localtime[<type>]		X	X	X	X	X	X	X	X	X	X	X
type ▲	utc *(default)*	X	X	X	X	X	X	X	X	X	X	X
	local	X	X	X	X	X	X	X	X	X	X	X
system.run[command,<mode>]		X	X	X	X	X	X	X	X	X	X	X
mode ▲	wait *(default)*	X	X	X	X	X	X	X	X	X	X	X
	nowait	X	X	X	X	X	X	X	X	X	X	X
system.stat[resource,<type>]		–	–	–	–	–	–	X	–	–	–	–
system.sw.arch		–	X	X	–	–	–	–	–	–	–	–
system.sw.os[<info>]		–	X	X	–	–	–	–	–	–	–	–
system.sw.packages[<package>,<manager>,<format>]		–	X	X	–	–	–	–	–	–	–	–
system.swap.in[<device>,<type>] *(specifying a device is only supported under Linux)*		–	X	X	–	X	–	–	–	–	X	–
type ▲ *(pages will only work if device was not specified)*	count *(default under all except Linux)*	–	X	X	–	X	–	–	–	–	X	–
	sectors	–	X	X	–	–	–	–	–	–	–	–
	pages *(default under Linux)*	–	X	X	–	X	–	–	–	–	X	–
system.swap.out[<device>,<type>] *(specifying a device is only supported under Linux)*		–	X	X	–	X	–	–	–	–	X	–
type ▲ *(pages will only work if device was not specified)*	count *(default under all except Linux)*	–	X	X	–	X	–	–	–	–	X	–
	sectors	–	X	X	–	–	–	–	–	–	–	–
	pages *(default under Linux)*	–	X	X	–	X	–	–	–	–	X	–
system.swap.size[<device>,<type>] *(specifying a device is only supported under Linux)*		X	X	X	X	X	–	–	X	–	X	–
type ▲	free *(default)*	X	X	X	X	X	–	–	X	–	X	–
	total	X	X	X	X	X	–	–	X	–	X	–
	used	–	X	X	X	–	–	–	–	–	X	–
	pfree	–	X	X	X	X	–	–	–	–	X	–
	pused	–	X	X	X	X	–	–	–	–	X	–
system.uname		X	X	X	X	X	X	X	X	X	X	X
system.uptime		X	X	X	X	X	–	X	?	X	X	X
system.users.num		–	X	X	X	X	X	X	X	X	X	X

		1	2	3	4	5	6	7	8	9	10	11
vfs.dev.read[<device>,<type>,<mode>]		–	X	X	X	X	–	–	–	–	X	–
type ▲ *(defaults are different under various OSes)*	sectors	–	X	X	–	–	–	–	–	–	–	–
	operations	–	X	X	X	X	–	–	–	–	X	–
	bytes	–	–	–	X	X	–	–	–	–	X	–
	sps	–	X	X	–	–	–	–	–	–	–	–
	ops	–	X	X	X	–	–	–	–	–	–	–
	bps	–	–	–	X	–	–	–	–	–	–	–
mode ▲ *(compatible only with type in: sps, ops, bps)*	avg1 *(default)*	–	X	X	X	–	–	–	–	–	i	–
	avg5	–	X	X	X	–	–	–	–	–	i	–
	avg15	–	X	X	X	–	–	–	–	–	i	–
vfs.dev.write[<device>,<type>,<mode>]		–	X	X	X	X	–	–	–	–	X	–
type ▲ *(defaults are different under various OSes)*	sectors	–	X	X	–	–	–	–	–	–	–	–
	operations	–	X	X	X	X	–	–	–	–	X	–
	bytes	–	–	–	X	X	–	–	–	–	X	–
	sps	–	X	X	–	–	–	–	–	–	–	–
	ops	–	X	X	X	–	–	–	–	–	–	–
	bps	–	–	–	X	–	–	–	–	–	–	–
mode ▲ *(compatible only with type in: sps, ops, bps)*	avg1 *(default)*	–	X	X	X	–	–	–	–	–	i	–
	avg5	–	X	X	X	–	–	–	–	–	i	–
	avg15	–	X	X	X	–	–	–	–	–	i	–
vfs.file.cksum[file]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.contents[file,<encoding>]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.exists[file]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.md5sum[file]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.regexp[file,regexp,<encoding>]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.regmatch[file,regexp,<encoding>]		X	X	X	X	X	X	X	X	X	X	X
vfs.file.size[file]		X	X	X	X	X	X	X	X	X	X	X
		1	2	3	4	5	6	7	8	9	10	11
vfs.file.time[file,<mode>]		X	X	X	X	X	X	X	X	X	X	X
mode ▲	modify *(default)*	X	X	X	X	X	X	X	X	X	X	X
	access	X	X	X	X	X	X	X	X	X	X	X
	change	X	X	X	X	X	X	X	X	X	X	X
vfs.fs.discovery		X	X	X	X	X	X	X	–	X	X	X
vfs.fs.inode[fs,<mode>]		–	X	X	X	X	X	X	X	X	X	X
mode ▲	total *(default)*	–	X	X	X	X	X	X	X	X	X	X
	free	–	X	X	X	X	X	X	X	X	X	X
	used	–	X	X	X	X	X	X	X	X	X	X
	pfree	–	X	X	X	X	X	X	X	X	X	X
	pused	–	X	X	X	X	X	X	X	X	X	X
vfs.fs.size[fs,<mode>]		X	X	X	X	X	X	X	X	X	X	X
	total *(default)*	X	X	X	X	X	X	X	X	X	X	X
	free	X	X	X	X	X	X	X	X	X	X	X

mode ▲	used	X	X	X	X	X	X	X	X	X	X	X
	pfree	X	X	X	X	X	X	X	X	X	X	X
	pused	X	X	X	X	X	X	X	X	X	X	X
vm.memory.size[<mode>]		X	X	X	X	X	X	X	X	X	X	X
mode ▲	total *(default)*	X	X	X	X	X	X	X	X	X	X	X
	active	–	–	–	X	–	X	–	–	X	X	X
	anon	–	–	–	–	–	–	–	–	–	–	X
	buffers	–	X	X	X	–	–	–	–	–	X	X
	cached	X	X	X	X	–	–	X	–	–	X	X
	exec	–	–	–	–	–	–	–	–	–	–	X
	file	–	–	–	–	–	–	–	–	–	–	X
	free	X	X	X	X	X	X	X	X	X	X	X
	inactive	–	–	–	X	–	–	–	–	X	X	X
	pinned	–	–	–	–	–	–	X	–	–	–	–
	shared	–	X	–	X	–	–	–	–	–	X	X
	wired	–	–	–	X	–	–	–	–	X	X	X
	used	X	X	X	X	X	X	X	X	X	X	X
	pused	X	X	X	X	X	X	X	X	X	X	X
	available	X	X	X	X	X	X	X	X	X	X	X
	pavailable	X	X	X	X	X	X	X	X	X	X	X
web.page.get[host,<path>,<port>]		X	X	X	X	X	X	X	X	X	X	X
web.page.perf[host,<path>,<port>]		X	X	X	X	X	X	X	X	X	X	X
web.page.regexp[host,<path>,<port>,<regexp>,<length>]		X	X	X	X	X	X	X	X	X	X	X
		1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11

See also a description of vm.memory.size parameters.

2.0/manual/appendix/items/supported_by_platform.txt · Last modified: 2013/04/12 15:36 by dimir
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

2 vm.memory.size parameters

- **total** - total physical memory available.
- **free** - memory that is readily available to any entity requesting memory.
- **active** - memory currently in use or very recently used, and so it is in RAM.
- **inactive** - memory that is marked as not used.
- **wired** - memory that is marked to always stay in RAM. It is never moved to disk.
- **pinned** - same as 'wired'.
- **anon** - memory not associated with a file (cannot be re-read from it).
- **exec** - executable code, typically from a (program) file.
- **file** - cache for contents of recently accessed files.
- **buffers** - cache for things like file system metadata.
- **cached** - cache for various things.
- **shared** - memory that may be simultaneously accessed by multiple processes.
- **used** - active + wired memory.
- **pused** - active + wired memory in relation to total.
- **available** - inactive + cached + free memory.
- **pavailable** - inactive + cached + free memory in relation to 'total'.

The sum of *vm.memory.size[used]* and *vm.memory.size[available]* does not necessarily equal total. For instance, on FreeBSD active, inactive, wired, cached memories are considered used, because they store some useful information.
At the same time inactive, cached, free memories are considered available, because these kinds of memories can be given instantly to processes that request more memory.
So inactive memory is both used and available simultaneously. Because of this, item *vm.memory.size[used]* is designed for informational purposes only, while item *vm.memory.size[available]* is designed to be used in triggers.

See the "See also" section at the bottom of this page to find more detailed information about memory calculation in different OS.

Platform-specific notes

- on AIX and Solaris **available** and **free** are the same
- on Linux **shared** works only on kernel 2.4

See also

1. Detailed information about memory calculation in different OS [http://blog.zabbix.com/when-alexei-isnt-looking#vm.memory.size]

2.0/manual/appendix/items/vm.memory.size_params.txt · Last modified: 2012/04/15 22:33 by zalex_ua
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

3 Passive and active agent checks

Overview

This section provides details on passive and active checks performed by Zabbix agent.

Zabbix uses a JSON based communication protocol for communicating with Zabbix agent.

There are some definitions used in the details of protocols used by Zabbix:

```
<HEADER> - "ZBXD\x01" (5 bytes)
<DATALEN> - data length (8 bytes). 1 will be formatted as 01/00/00/00/00/00/00/00 (eight bytes in HEX, 64 bit number)
```

To not exhaust memory (potentially) Zabbix server is limited to accept only 128MB in one connection (unlimited before Zabbix 2.0.3) when using the Zabbix protocol.

Passive checks

A passive check is a simple data request. Zabbix server or proxy asks for some data (for example, CPU load) and Zabbix agent sends back the result to the server.

Server request

```
<item key>\n
```

Agent response

```
<HEADER><DATALEN><DATA>
```

For example:

1. Server opens a TCP connection
2. Server sends **agent.ping\n**
3. Agent reads the request and responds with **<HEADER><DATALEN>1**
4. Server processes data to get the value, '1' in our case
5. TCP connection is closed

Active checks

Active checks require more complex processing. The agent must first retrieve from the server(s) a list of items for independent processing.

The servers to get the active checks from are listed in the 'ServerActive' parameter of the agent configuration file. The frequency of asking for these checks is set by the 'RefreshActiveChecks' parameter in the same configuration file. However, if refreshing active checks fails, it is retried after hardcoded 60 seconds.

The agent then periodically sends the new values to the server(s).

Getting the list of items

Agent request

```
<HEADER><DATALEN>{
   "request":"active checks",
   "host":"<hostname>"
```

```
}
```

Server response

```
{
    "response":"success",
    "data":[
    {
        "key":"log[\/home\/zabbix\/logs\/zabbix_agentd.log]",
        "delay":"30",
        "lastlogsize":"0"
    },
    {
        "key":"agent.version",
        "delay":"600"
    }
    ]
}
```

The server must respond with success. For each returned item, **key** and **delay** must exist. For items having type “Log”, the **lastlogsize** must exist as well.

If sending of some values fails on the server (for example, because host or item has been disabled or deleted), agent will not retry sending of those values.

For example:

1. Agent opens a TCP connection
2. Agent asks for the list of checks
3. Server responds with a list of items (item key, delay)
4. Agent parses the response
5. TCP connection is closed
6. Agent starts periodical collection of data

Sending in collected data

Agent sends

```
<HEADER><DATALEN>{
   "request":"agent data",
   "data":[
       {
           "host":"<hostname>",
           "key":"log[\/home\/zabbix\/logs\/zabbix_agentd.log]",
           "value":" 13039:20090907:184546.759 zabbix_agentd started. ZABBIX 1.6.6 (revision {7836}).",
           "lastlogsize":80,
           "clock":1252926015
       },
       {
           "host":"<hostname>",
           "key":"agent.version",
           "value":"1.6.6",
           "clock":1252926015
       }
   ],
   "clock":1252926016
}
```

Server response

```
<HEADER><DATALEN>{
    "response":"success",
    "info":"Processed 2 Failed 0 Total 2 Seconds spent 0.002070"
```

```
}
```

For example:

1. Agent opens a TCP connection
2. Agent sends a list of values
3. Server processes the data and sends the status back
4. TCP connection is closed

Older XML protocol

Zabbix server will take up to 16 MB of XML Base64-encoded data in one connection, but a single decoded value should be no longer than 64 KB otherwise it will be truncated to 64 KB while decoding.

2.0/manual/appendix/items/activepassive.txt · Last modified: 2013/05/14 16:57 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

4 Encoding of returned values

Zabbix server expects every returned text value in the UTF8 encoding. This is related to any type of checks: zabbix agent, ssh, telnet, etc.

Different monitored systems/devices and checks can return non-ASCII characters in the value. For such cases, almost all possible zabbix keys contain an additional item key parameter - <**encoding**>. This key parameter is optional but it should be specified if the returned value is not in the UTF8 encoding and it contains non-ASCII characters. Otherwise the result can be unexpected and unpredictable.

A description of behavior with different database back-ends in such cases follows.

MySQL

If a value contains a non-ASCII character in non UTF8 encoding - this character and the following will be discarded when the database stores this value. No warning messages will be written to the *zabbix_server.log*.
Relevant for at least MySQL version 5.1.61

PostgreSQL

If a value contains a non-ASCII character in non UTF8 encoding - this will lead to a failed SQL query (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) and data will not be stored. An appropriate warning message will be written to the *zabbix_server.log*.
Relevant for at least PostgreSQL version 9.1.3

2.0/manual/appendix/items/encoding_of_values.txt · Last modified: 2012/04/12 21:23 by zalex_ua
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

5 Large file support

Large file support, often abbreviated to LFS, is the term applied to the ability to work with files larger than 2 GB on 32-bit operating systems. Since Zabbix 2.0 support for large files has been added. This change affects at least log file monitoring and all vfs.file.* items. Large file support depends on the capabilities of a system at Zabbix compilation time, but is completely disabled on a 32-bit Solaris due to its incompatibility with procfs and swapctl.

2.0/manual/appendix/items/large_file_support.txt · Last modified: 2012/05/21 13:18 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

6 Unreachable/unavailable host settings

Overview

Several configuration parameters define how Zabbix server should behave when an agent check (Zabbix, SNMP, IPMI, JMX) fails and a host becomes unreachable.

Unreachable host

A host is treated as unreachable after a failed agent check (network error, timeout).

From that moment **UnreachableDelay** defines how often items are rechecked in this unreachability situation and such rechecks will be performed already by unreachable pollers. By default it is 15 seconds before the next check.

In the Zabbix server log unreachability is indicated by messages like these:

```
Zabbix agent item [system.cpu.load[percpu,avg1]] on host [New host] failed: first network error, wait for 15 seconds
Zabbix agent item [system.cpu.load[percpu,avg15]] on host [New host] failed: another network error, wait for 15 seconds
```

Note that the exact item that failed is indicated and the item type (Zabbix agent).

The *Timeout* parameter will also affect how early an item is rechecked during unreachability. If the Timeout is 20 seconds and UnreachableDelay 30 seconds, the next check will be in 50 seconds after the first attempt.

The **UnreachablePeriod** parameter defines how long the unreachability period is in total. By default UnreachablePeriod is 45 seconds. UnreachablePeriod should be several times bigger than UnreachableDelay, so that items are rechecked more than once before a host becomes unavailable.

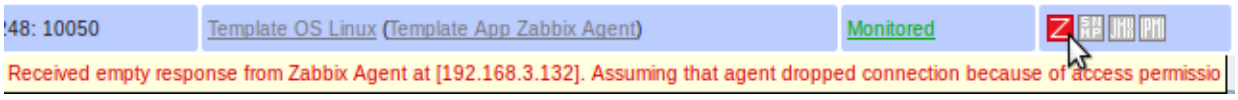
Unavailable host

After the UnreachablePeriod ends and the host has not reappeared, the host is treated as unavailable.

In the server log it is indicated by messages like these:

```
temporarily disabling Zabbix agent checks on host [New host]: host unavailable
```

and in the frontend the host availability icon goes from green to red (note that on mouseover a tooltip with the error description is displayed):



The **UnavailableDelay** parameter defines how often items are checked during host unavailability.

By default it is 60 seconds (so in this case “temporarily disabling”, from the log message above, will mean disabling checks for one minute).

When the connection to the host is restored, the monitoring returns to normal automatically:

```
enabling Zabbix agent checks on host [New host]: host became available
```

2.0/manual/appendix/items/unreachability.txt · Last modified: 2013/09/23 10:55 by martins-v
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

5 Triggers

2.0/manual/appendix/triggers.txt · Last modified: 2012/02/20 15:14 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Supported trigger functions

All functions supported in trigger expressions are listed here:

▼	FUNCTION	*Parameter(s)*	*Supported value types*
	Definition		
	abschange	***ignored***	***float, int, str, text, log***
	Returns absolute difference between last and previous values. For strings: 0 – values are equal 1 – values differ		
	avg	***sec or #num***	***float, int***
	The average value of a period. The period is set in seconds or as a number of collected values (preceded by a # hash mark, for example, *#5* – five latest values) in the first parameter. The function accepts a second, optional parameter **time_shift**. It is useful when there is a need to compare the current average value with the average value time_shift seconds back. For instance, avg(3600,86400) will return the average value for an hour one day ago. Parameter time_shift is supported from Zabbix 1.8.2.		
	change	***ignored***	***float, int, str, text, log***
	Returns difference between last and previous values. For strings: 0 – values are equal 1 – values differ		
	count	***sec or #num***	***float, int, str, text, log***
	The number of collected values in a period. The period is set in seconds or as a number of values (preceded by a # hash mark) in the first parameter. The function accepts a second optional **pattern** parameter, a third **operator** parameter, and a fourth **time_shift** parameter. For *pattern*: integer items – exact match; float items – match within 0.000001 Supported *operators* *: **eq** – equal **ne** – not equal **gt** – greater **ge** – greater or equal **lt** – less **le** – less or equal **like** – matches if contains pattern. *For integer and float items: operators **eq** (default), **ne**, **gt**, **ge**, **lt**, **le** are supported. *For string, text and log items: operators **like** (default), **eq**, **ne** are supported. Examples: *count(600)* – the number of values for last 10 minutes *count(600,12)* – the number of values for last 10 minutes that equal '12' *count(600,12,"gt")* – the number of values for last 10 minutes that are over '12' *count(#10,12,"gt")* – the number of values within last 10 values that are over '12' *count(600,12,"gt",86400)* – the number of values for preceding 10 minutes up to 24 hours ago that were over '12' *count(600,,,86400)* – the number of values for preceding 10 minutes up to 24 hours ago **#num** parameter is supported from Zabbix 1.6.1. **time_shift** parameter and string operators are supported from Zabbix 1.8.2. See function avg() for an example of using time_shift.		
	date	***ignored***	***any***
	Returns current date in YYYYMMDD format. For example: 20031025		
	dayofmonth	***ignored***	***any***

	Returns day of month in range of 1 to 31. This function is supported since Zabbix 1.8.5.	
dayofweek	***ignored***	***any***
	Returns day of week in range of 1 to 7. Mon - 1, Sun - 7.	
delta	***sec or #num***	***float, int***
	Returns the difference between the maximum and minimum values in a period ('max()' minus 'min()'). The period is set in seconds or as a number of collected values (preceded by a # hash mark) in the first parameter. Since Zabbix 1.8.2, the function supports a second, optional parameter time_shift. See function avg() for an example of its use.	
diff	***ignored***	***float, int, str, text, log***
	Returns: 1 - last and previous values differ 0 - otherwise	
fuzzytime	***sec***	***float, int***
	Returns 1 if timestamp (item value) does not differ from Zabbix server time for more than N seconds, 0 - otherwise. Usually used with system.localtime to check that local time is in sync with local time of Zabbix server.	
iregexp	***1st - string, 2nd - sec or #num***	***str, log, text***
	This function is non case-sensitive analogue of **regexp**.	
last	***sec or #num***	***float, int, str, text, log***
	Last (most recent) value. Parameter: *sec* - ignored *#num* - Nth most recent value Take note that **#num**, as can be expected, works differently here than with many other functions. For example, last(0) is always equal to last(#1) last(#3) - third most recent value (*not* three latest values) The function also supports a second optional **time_shift** parameter. For example, last(0,86400) will return the most recent value one day ago. Zabbix does not guarantee exact order of values if more than two values exist within one second in history. Parameter #num is supported starting from Zabbix 1.6.2. Parameter time_shift is supported starting from Zabbix 1.8.2. See function avg() for an example of its use.	
logeventid	***string***	***log***
	Check if Event ID of the last log entry matches a regular expression. Parameter defines the regular expression, POSIX extended [http://en.wikipedia.org/wiki/Regular_expression#POSIX_Extended_Regular_Expressions] style. Returns: 0 - does not match 1 - matches This function is supported since Zabbix 1.8.5.	
logseverity	***ignored***	***log***
	Returns log severity of the last log entry. Parameter is ignored. 0 - default severity N - severity (integer, useful for Windows event logs). Zabbix takes log severity from field **Information** of Windows event log.	
logsource	***string***	***log***
	Check if log source of the last log entry matches parameter. 0 - does not match 1 - matches Normally used for Windows event logs. For example, logsource["VMWare Server"]	
max	***sec or #num***	***float, int***
	The maximum (highest) value of a period. The period is set in seconds or as a number of collected values (preceded by a #	

	hash mark) in the first parameter. Since Zabbix 1.8.2, the function supports a second, optional parameter time_shift. See function avg for an example of its use.	
min	***sec or #num***	***float, int***
	The minimum (lowest) value of a period. The period is set in seconds or as a number of collected values (preceded by a # hash mark) in the first parameter. Since Zabbix 1.8.2, the function supports a second, optional parameter time_shift. See function avg for an example of its use.	
nodata	***sec***	***any***
	Returns: 1 - if no data received during period of time in seconds. The period should not be less than 30 seconds. 0 - otherwise	
now	***ignored***	***any***
	Returns number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).	
prev	***ignored***	***float, int, str, text, log***
	Returns previous value. Parameter is ignored. Same as last(#2)	
regexp	***1st - string, 2nd - sec or #num***	***str, log, text***
	Check if last (most recent) value matches regular expression. Parameter defines regular expression, POSIX extended [http://en.wikipedia.org/wiki/Regular_expression#POSIX_Extended_Regular_Expressions] style. Second optional parameter is number of seconds or number of collected values to analyse. In this case more than one value will be processed. This function is case-sensitive. Returns: 1 - found 0 - otherwise	
str	***1st - string, 2nd - sec or #num***	***str, log, text***
	Find string in last (most recent) value. Parameter defines string to find. Case sensitive! Second optional parameter is number of seconds or number of collected values to analyse. In this case more than one value will be processed. Returns: 1 - found 0 - otherwise	
strlen	***sec or #num***	***str, log, text***
	Length of the last (most recent) value in characters (not bytes). Parameters work the same as with 'last' function (see above in the table). For example, strlen(0) is equal to strlen(#1) strlen(#3) - length of the third most recent value strlen(0,86400) - length of the most recent value one day ago. This function is supported since Zabbix 1.8.4.	
sum	***sec or #num***	***float, int***
	The sum of collected values in a period. The period is set in seconds or as a number of values (preceded by a # hash mark) in the first parameter. Since Zabbix 1.8.2, the function supports a second, optional parameter time_shift. See function avg for an example of its use.	
time	***ignored***	***any***
	Returns current time in HHMMSS format. Example: 123055	

1) All functions return numeric values only. Comparison to strings is not supported, for example.
2) Some of the functions cannot be used for non-numeric parameters!
3) String arguments should be double quoted. Otherwise, they might get misinterpreted.

2.0/manual/appendix/triggers/functions.txt · Last modified: 2013/05/22 10:51 by martins-v

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

6 Macros

2.0/manual/appendix/macros.txt · Last modified: 2012/04/15 21:35 by zalex_ua

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

1 Macros supported by location

Overview

The table contains a complete list of macros supported by Zabbix:

- **X** means “supported” in that location
- Macros with numbers <**1-9**> like {MACRO**1**}, {MACRO**2**}…{MACRO**9**} allow to reference hosts in the order in which they appear in a trigger expression. So, {HOST.IP**1**} and {HOST.IP**2**} will expand to the IPs of the first and second host in the trigger expression.

DB monitor additional parameters, SSH and Telnet scripts													DESCRIPTION
Interface IP/DNS												▼▼	
Item names											▼▼		
Trigger names										▼▼			
Trigger expressions									▼▼				
Map URLs								▼▼					
Map labels[1]							▼▼						
Item key's parameters						▼▼							
GUI Scripts					▼▼								
Auto registration notifications				▼▼									
Discovery notifications			▼▼										
Notifications and commands		▼▼											
MACRO	▼▼												
▼▼	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{DATE}	X	X	X										Current date in yyyy.mm.dd. format.
{DISCOVERY.DEVICE.IPADDRESS}		X											IP address of the discovered device. Available always, does not depend on host being added.
{DISCOVERY.DEVICE.DNS}		X											DNS name of the discovered device. Available always, does not depend on host being added.
{DISCOVERY.DEVICE.STATUS}		X											Status of the discovered device: can be either UP or DOWN.
{DISCOVERY.DEVICE.UPTIME}		X											Time since the last change of discovery status for a particular device. For example: 1h 29m. For devices with status DOWN, this is the period of their downtime.
{DISCOVERY.RULE.NAME}		X											Name of the discovery rule that discovered the presence or absence of the device or service.
{DISCOVERY.SERVICE.NAME}		X											Name of the service that was discovered. For example: HTTP.

{DISCOVERY.SERVICE.PORT}		X											Port of the service that was discovered. For example: 80.
{DISCOVERY.SERVICE.STATUS}		X											Status of the discovered service: can be either UP or DOWN.
{DISCOVERY.SERVICE.UPTIME}		X											Time since the last change of discovery status for a particular service. For example: 1h 29m. For services with status DOWN, this is the period of their downtime.
{ESC.HISTORY}	X												Escalation history. Log of previously sent messages. Shows previously sent notifications, on which escalation step they were sent and their status (*sent*, *in progress* or *failed*).
{EVENT.ACK.HISTORY}	X												
{EVENT.ACK.STATUS}	X												
{EVENT.AGE}	X	X	X										Age of the event. Useful in escalated messages.
{EVENT.DATE}	X	X	X										Date of the event.
{EVENT.ID}	X	X	X										Numeric event ID which triggered this action.
{EVENT.TIME}	X	X	X										Time of the event.
{HOST.CONN<1-9>}	X			X	X[2]	X			X		X	X[5]	IP or host DNS name depending on host settings[3]. Supported in trigger names since 2.0.0.
{HOST.DNS<1-9>}	X			X	X[2]	X			X		X	X[5]	Host DNS name[3]. Supported in trigger names since 2.0.0.
{HOST.HOST<1-9>}	X		X	X	X[2]	X			X		X	X[5]	Host name. **{HOSTNAME<1-9>} is deprecated.**
{HOST.ID}							X						Host ID.
{HOST.IP<1-9>}	X		X	X	X[2]	X			X		X	X[5]	Host IP address[3]. Supported since 2.0.0. **{IPADDRESS<1-9>} is deprecated.**
{HOST.NAME<1-9>}	X		X	X	X[2]	X			X		X	X[5]	Visible host name. Supported since 2.0.0.
{HOST.PORT}			X										Host (agent) port[3]. Supported since 2.0.0.
{HOSTGROUP.ID}							X						Host group ID.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{INVENTORY.ALIAS<1-9>}	X												Alias field in host inventory.
{INVENTORY.ASSET.TAG<1-9>}	X												Asset tag field in host inventory.
{INVENTORY.CHASSIS<1-9>}	X												Chassis field in host inventory.
{INVENTORY.CONTACT<1-9>}	X												Contact field in host inventory. **{PROFILE.CONTACT<1-9>} is deprecated.**

{INVENTORY.CONTRACT.NUMBER<1-9>}	X												Contract number field in host inventory.
{INVENTORY.DEPLOYMENT.STATUS<1-9>}	X												Deployment status field in host inventory.
{INVENTORY.HARDWARE<1-9>}	X												Hardware field in host inventory. **{PROFILE.HARDWARE<1-9>} is deprecated.**
{INVENTORY.HARDWARE.FULL<1-9>}	X												Hardware (Full details) field in host inventory.
{INVENTORY.HOST.NETMASK<1-9>}	X												Host subnet mask field in host inventory.
{INVENTORY.HOST.NETWORKS<1-9>}	X												Host networks field in host inventory.
{INVENTORY.HOST.ROUTER<1-9>}	X												Host router field in host inventory.
{INVENTORY.HW.ARCH<1-9>}	X												Hardware architecture field in host inventory.
{INVENTORY.HW.DATE.DECOMM<1-9>}	X												Date hardware decommissioned field in host inventory.
{INVENTORY.HW.DATE.EXPIRY<1-9>}	X												Date hardware maintenance expires field in host inventory.
{INVENTORY.HW.DATE.INSTALL<1-9>}	X												Date hardware installed field in host inventory.
{INVENTORY.HW.DATE.PURCHASE<1-9>}	X												Date hardware purchased field in host inventory.
{INVENTORY.INSTALLER.NAME<1-9>}	X												Installer name field in host inventory.
{INVENTORY.LOCATION<1-9>}	X												Location field in host inventory. **{PROFILE.LOCATION<1-9>} is deprecated.**
{INVENTORY.LOCATION.LAT<1-9>}	X												Location latitude field in host inventory.
{INVENTORY.LOCATION.LON<1-9>}	X												Location longitude field in host inventory.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{INVENTORY.MACADDRESS.A<1-9>}	X												MAC address A field in host inventory. **{PROFILE.MACADDRESS<1-9>} is deprecated.**
{INVENTORY.MACADDRESS.B<1-9>}	X												MAC address B field in host inventory.
{INVENTORY.MODEL<1-9>}	X												Model field in host inventory.
{INVENTORY.NAME<1-9>}	X												Name field in host inventory. **{PROFILE.NAME<1-9>} is deprecated.**
{INVENTORY.NOTES<1-9>}	X												Notes field in host inventory. **{PROFILE.NOTES<1-9>} is deprecated.**
{INVENTORY.OOB.IP<1-9>}	X												OOB IP address field in host inventory.
{INVENTORY.OOB.NETMASK<1-9>}	X												OOB subnet mask field in host inventory.

{INVENTORY.OOB.ROUTER<1-9>}	X												OOB router field in host inventory.
{INVENTORY.OS<1-9>}	X												OS field in host inventory. **{PROFILE.OS<1-9>} is deprecated.**
{INVENTORY.OS.FULL<1-9>}	X												OS (Full details) field in host inventory.
{INVENTORY.OS.SHORT<1-9>}	X												OS (Short) field in host inventory.
{INVENTORY.POC.PRIMARY.CELL<1-9>}	X												Primary POC cell field in host inventory.
{INVENTORY.POC.PRIMARY.EMAIL<1-9>}	X												Primary POC email field in host inventory.
{INVENTORY.POC.PRIMARY.NAME<1-9>}	X												Primary POC name field in host inventory.
{INVENTORY.POC.PRIMARY.NOTES<1-9>}	X												Primary POC notes field in host inventory.
{INVENTORY.POC.PRIMARY.PHONE.A<1-9>}	X												Primary POC phone A field in host inventory.
{INVENTORY.POC.PRIMARY.PHONE.B<1-9>}	X												Primary POC phone B field in host inventory.
{INVENTORY.POC.PRIMARY.SCREEN<1-9>}	X												Primary POC screen name field in host inventory.
{INVENTORY.POC.SECONDARY.CELL<1-9>}	X												Secondary POC cell field in host inventory.
{INVENTORY.POC.SECONDARY.EMAIL<1-9>}	X												Secondary POC email field in host inventory.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{INVENTORY.POC.SECONDARY.NAME<1-9>}	X												Secondary POC name field in host inventory.
{INVENTORY.POC.SECONDARY.NOTES<1-9>}	X												Secondary POC notes field in host inventory.
{INVENTORY.POC.SECONDARY.PHONE.A<1-9>}	X												Secondary POC phone A field in host inventory.
{INVENTORY.POC.SECONDARY.PHONE.B<1-9>}	X												Secondary POC phone B field in host inventory.
{INVENTORY.POC.SECONDARY.SCREEN<1-9>}	X												Secondary POC screen name field in host inventory.
{INVENTORY.SERIALNO.A<1-9>}	X												Serial number A field in host inventory. **{PROFILE.SERIALNO<1-9>} is deprecated.**
{INVENTORY.SERIALNO.B<1-9>}	X												Serial number B field in host inventory.
{INVENTORY.SITE.ADDRESS.A<1-9>}	X												Site address A field in host inventory.
{INVENTORY.SITE.ADDRESS.B<1-9>}	X												Site address B field in host inventory.
{INVENTORY.SITE.ADDRESS.C<1-9>}	X												Site address C field in host inventory.
{INVENTORY.SITE.CITY<1-9>}	X												Site city field in host inventory.
{INVENTORY.SITE.COUNTRY<1-9>}	X												Site country field in host inventory.
{INVENTORY.SITE.NOTES<1-9>}	X												Site notes field in host

{INVENTORY.[illegible]}	X												inventory.
{INVENTORY.SITE.RACK<1-9>}	X												Site rack location field in host inventory.
{INVENTORY.SITE.STATE<1-9>}	X												Site state / province field in host inventory.
{INVENTORY.SITE.ZIP<1-9>}	X												Site ZIP / postal field in host inventory.
{INVENTORY.SOFTWARE<1-9>}	X												Software field in host inventory. **{PROFILE.SOFTWARE<1-9>} is deprecated.**
{INVENTORY.SOFTWARE.APP.A<1-9>}	X												Software application A field in host inventory.
{INVENTORY.SOFTWARE.APP.B<1-9>}	X												Software application B field in host inventory.
{INVENTORY.SOFTWARE.APP.C<1-9>}	X												Software application C field in host inventory.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{INVENTORY.SOFTWARE.APP.D<1-9>}	X												Software application D field in host inventory.
{INVENTORY.SOFTWARE.APP.E<1-9>}	X												Software application E field in host inventory.
{INVENTORY.SOFTWARE.FULL<1-9>}	X												Software (Full details) field in host inventory.
{INVENTORY.TAG<1-9>}	X												Tag field in host inventory. **{PROFILE.TAG<1-9>} is deprecated.**
{INVENTORY.TYPE<1-9>}	X												Type field in host inventory. **{PROFILE.DEVICETYPE<1-9>} is deprecated.**
{INVENTORY.TYPE.FULL<1-9>}	X												Type (Full details) field in host inventory.
{INVENTORY.URL.A<1-9>}	X												URL A field in host inventory.
{INVENTORY.URL.B<1-9>}	X												URL B field in host inventory.
{INVENTORY.URL.C<1-9>}	X												URL C field in host inventory.
{INVENTORY.VENDOR<1-9>}	X												Vendor field in host inventory.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{ITEM.ID<1-9>}	X												Numeric ID of the Nth item in the trigger expression that caused a notification. Supported since 1.8.12.
{ITEM.DESCRIPTION<1-9>}	X												Description of the Nth item in the trigger expression that caused a notification. Supported since 2.0.0.
{ITEM.KEY<1-9>}	X												Key of the Nth item in the trigger expression that caused a notification. Supported since 2.0.0. **{TRIGGER.KEY} is deprecated.**
{ITEM.KEY.ORIG<1-9>}	X												Original key (with macros not expanded) of the Nth item in the trigger

													expression that caused a notification. Supported since 2.0.6.
{ITEM.LASTVALUE<1-9>}	X								X				The latest value of the Nth item in the trigger expression that caused a notification. Supported since 1.4.3. It is alias to {{HOST.HOST}: {ITEM.KEY}.last(0)}
{ITEM.LOG.AGE<1-9>}	X												
{ITEM.LOG.DATE<1-9>}	X												
{ITEM.LOG.EVENTID<1-9>}	X												
{ITEM.LOG.NSEVERITY<1-9>}	X												
{ITEM.LOG.SEVERITY<1-9>}	X												
{ITEM.LOG.SOURCE<1-9>}	X												
{ITEM.LOG.TIME<1-9>}	X												
{ITEM.NAME<1-9>}	X												Name of the Nth item in the trigger expression that caused a notification.
{ITEM.NAME.ORIG<1-9>}	X												Original name (with macros not expanded) of the Nth item in the trigger expression that caused a notification. Supported since 2.0.6.
{ITEM.VALUE<1-9>}	X								X				Resolved to either: 1) the latest value of the Nth item in the trigger expression, if used for displaying triggers. In this case, works the same as {ITEM.LASTVALUE}. 2) the historical (at-the-time-of-event) value of the Nth item in the trigger expression, if used for displaying events and notifications. Will resolve to UNKNOWN if the history value has already been deleted or has never been stored. Supported since 1.4.3.
{MAP.ID}							X						Network map ID.
{NODE.ID<1-9>}	X	X	X										
{NODE.NAME<1-9>}	X	X	X										
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{PROXY.NAME<1-9>}	X	X	X										Proxy name of the Nth item in the trigger expression that caused a notification. Supported since 1.8.4.
{TIME}	X	X	X										Current time in hh:mm.ss.
{TRIGGER.DESCRIPTION}	X												Trigger description. Supported since 2.0.4. **{TRIGGER.COMMENT} is deprecated.**
{TRIGGER.EVENTS.ACK}	X					X							Number of acknowledged events for a map element in maps, or for the trigger

{TRIGGER.EVENTS.ACK}	X					X							which generated current event in notifications. Supported since 1.8.3.
{TRIGGER.EVENTS.PROBLEM.ACK}	X					X							Number of acknowledged PROBLEM events for all triggers disregarding their state. Supported since 1.8.3.
{TRIGGER.EVENTS.PROBLEM.UNACK}	X					X							Number of unacknowledged PROBLEM events for all triggers disregarding their state. Supported since 1.8.3.
{TRIGGER.EVENTS.UNACK}	X					X							Number of unacknowledged events for a map element in maps, or for the trigger which generated current event in notifications. Supported in map element labels since 1.8.3.
{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}						X							Number of acknowledged PROBLEM events for triggers in PROBLEM state. Supported since 1.8.3.
{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}						X							Number of unacknowledged PROBLEM events for triggers in PROBLEM state. Supported since 1.8.3.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	
{TRIGGER.EXPRESSION}	X												Trigger expression. Supported since 1.8.12.
{TRIGGER.HOSTGROUP.NAME}	X												Sorted (by SQL query) comma-space separated list of host groups in which the trigger is defined. Supported since 2.0.6.
{TRIGGER.ID}	X						X						Numeric trigger ID which triggered this action.
{TRIGGER.NAME}	X												Name of the trigger.
{TRIGGER.NAME.ORIG}	X												Original name (with macros not expanded) of the trigger. Supported since 2.0.6.
{TRIGGER.NSEVERITY}	X												Numerical trigger severity. Possible values: 0 - Not classified, 1 - Information, 2 - Warning, 3 - Average, 4 - High, 5 - Disaster. Supported starting from Zabbix 1.6.2.
{TRIGGER.SEVERITY}	X												Trigger severity name. Can be defined in *Administration → General → Trigger severities*.
{TRIGGER.STATUS}	X												Trigger state. Can be either PROBLEM or OK. **{STATUS} is deprecated.**
{TRIGGER.TEMPLATE.NAME}	X												Sorted (by SQL query) comma-space separated list of templates in which the trigger is defined or *UNKNOWN* if the trigger is defined in a host. Supported since 2.0.6

													since 2.0.6.
{TRIGGER.URL}	X												Trigger URL.
{TRIGGER.VALUE}	X							X					Current trigger value: 0 - trigger is in OK state, 1 - trigger is in PROBLEM state, 2 - trigger UNKNOWN. This macro can also be used in trigger expressions.
{TRIGGERS.UNACK}						X							Number of unacknowledged triggers for a map element, disregarding trigger state. Trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged.
{TRIGGERS.PROBLEM.UNACK}						X							Number of unacknowledged PROBLEM triggers for a map element. Trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged. Supported since 1.8.3.
{TRIGGERS.ACK}						X							Number of acknowledged triggers for a map element, disregarding trigger state. Trigger is considered to be acknowledged if all of it's PROBLEM events are acknowledged. Supported since 1.8.3.
{TRIGGERS.PROBLEM.ACK}						X							Number of acknowledged PROBLEM triggers for a map element. Trigger is considered to be acknowledged if all of it's PROBLEM events are acknowledged. Supported since 1.8.3.
{host:key.func(param)}	X					X[4]		X					Simple macros as used in trigger expressions.
{$MACRO}					X			X	X	X	X	X	User macros. Supported in item and trigger names since 1.8.4.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	

[1] Macros for map labels are supported since 1.8.

[2] Since 2.0.3 HOST.* macros supported in item keys will only work for item types that have interfaces. E. g. they will not work for types “Zabbix agent (active)”, “Calculated” etc.

[3] Depending on the context, the macro will be expanded differently.

In 2.0.1 or earlier version in the context of remote command execution, GUI scripts, item key parameters, and interface IP/DNS fields only the main agent interface will be considered as the source of information.

Since 2.0.2 the macro will use the main agent interface, however, if it is not present, main SNMP interface will be used. If SNMP is also not present, main JMX interface will be used. If JMX is not present either, main IPMI interface will be used.

In item key parameters the interface that is selected for the item will be used, since 2.0.3.

[4] Only functions **last**, **avg**, **max** and **min** with seconds as arguments are supported in map labels.

[5] Supported since 2.0.3.

Macro {TRIGGER.ID} is supported in trigger URL since Zabbix 1.8.8.

Additional support for user macros

In addition to the locations listed, user definable macros since Zabbix 2.0 are supported in numerous other locations.

- Hosts
  - Interface IP/DNS
  - Interface port
- Passive proxy
  - Interface port
- Items
  - SNMPv3 security name
  - SNMPv3 auth pass
  - SNMPv3 priv pass
  - SNMPv1/v2 community
  - SNMP OID
  - DB monitor additional parameters
  - SSH username
  - SSH public key
  - SSH private key
  - SSH password
  - SSH script
  - Telnet username
  - Telnet password
  - Telnet script
  - Calculated item formula
- Discovery
  - SNMPv3 security name
  - SNMPv3 auth pass
  - SNMPv3 priv pass
  - SNMPv1/v2 community
  - SNMP OID

Macros used in low-level discovery

There is a type of macro used within the low-level discovery function - **{#MACRO}**. It is a macro that is used in an LLD rule and returns real values of file system names, network interfaces and SNMP OIDs.

These macros can be used for creating item, trigger and graph *prototypes*. Then, when discovering real file systems, network interfaces etc., these macros are substituted with real values and are the basis for creating real items, triggers and graphs.

LLD macros can be used:

- for item prototypes in
  - names
  - key parameters
  - SNMP OIDs
  - calculated item formulas
  - SSH and Telnet scripts
  - database monitor item parameters
- for trigger prototypes in
  - names
  - expressions (insofar as when referencing an item key prototype)
- for graph prototypes in
  - names

Some low-level discovery macros come “pre-packaged” with the LLD funtion in Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. However, adhering to these names is not compulsory when creating a custom low-level discovery rule. Then you may use any other LLD macro name and refer to that name.

2.0/manual/appendix/macros/supported_by_location.txt · Last modified: 2013/10/03 12:33 by Pavels
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

7 Setting time periods

1 Format

To set a time period, the following format has to be used:

```
d-d,hh:mm-hh:mm
```

You can specify more than one time period using a semicolon (;) separator:

```
d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...
```

2 Description

Symbol	Description
d	Day of the week: 1 - Monday, 2 - Tuesday ,... , 7 - Sunday
hh	Hours: 00-24
mm	Minutes: 00-59

3 Default

Empty time period specification equals 01-07,00:00-24:00, which is the default value.

The upper limit of a time period is not included. Thus, if you specify 09:00-18:00 the last second included in the time period is 17:59:59. This is true starting from version 1.8.7, for everything, while Working time has always worked this way.

4 Examples

Working hours. Monday - Friday from 9:00 till 18:00:

```
1-5,09:00-18:00
```

Working hours plus weekend. Monday - Friday from 9:00 till 18:00 and Saturday, Sunday from 10:00 till 16:00:

```
1-5,09:00-18:00;6-7,10:00-16:00
```

2.0/manual/appendix/time_period.txt · Last modified: 2012/09/02 21:53 by dotneft
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

8 Command execution

Zabbix uses common functionality to execute user parameters, remote commands, system.run[] items without "nowait" flag, scripts (alert, external and global) and some internal commands. The command/script is executed similarly on both UNIX and Windows platforms:

1. Zabbix (the parent process) creates a pipe for communication
2. Zabbix sets the pipe as the output for the to-be-created child process
3. Zabbix creates the child process (runs the command/script)
4. A new process group (in UNIX) or a job (in Windows) is created for the child process
5. Zabbix reads from the pipe till timeout occurs or no one is writing to the other end (ALL handles/file descriptors have been closed). Note that the child process can create more processes and exit before they exit or close the handle/file descriptor.
6. If the timeout has not been reached, Zabbix waits till the initial child process exits or timeout occurs
7. At this point it is assumed that everything is done and the whole process tree (i.e. the process group or the job) is terminated

Please note that Zabbix assumes that a command/script has done processing when the initial child process has exited AND no other process is still keeping the output handle/file descriptor open. When processing is done, ALL created processes are terminated.

Read more about user parameters, remote commands, alert scripts.

All double quotes and backslashes in the command are escaped with backslashes and the command is enclosed in double quotes.

2.0/manual/appendix/command_execution.txt · Last modified: 2013/05/07 13:29 by richlv
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

9 Recipes for monitoring

General

Monitoring server availability

At least three methods (or combination of all methods) may be used in order to monitor availability of a server.

- ICMP ping (“icmpping” key)
- “zabbix[host,agent,available]” item
- trigger function nodata() for monitoring the availability of hosts that use active checks only

Sending alerts via WinPopUps

WinPopUps maybe very useful if you're running Windows OS and want to get quick notification from Zabbix. It could be good addition for email-based alert messages. Details about enabling of WinPopUps can be found at http://www.zabbix.com/forum/showthread.php?t=2147 [http://www.zabbix.com/forum/showthread.php?t=2147].

Monitoring specific applications

AS/400

IBM AS/400 platform can be monitored using SNMP. More information is available at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg244504.html?Open [http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg244504.html?Open].

MySQL

Several user parameters can be used for the monitoring of MySQL in the agent configuration file: /usr/local/etc/zabbix_agentd.conf

```
### Set of parameters for monitoring MySQL server (v3.23.42 and later)
### Change -u and add -p if required
#UserParameter=mysql.ping,mysqladmin -uroot  ping|grep alive|wc -l
#UserParameter=mysql.uptime,mysqladmin -uroot status|cut -f2 -d":"|cut -f2 -d" "
#UserParameter=mysql.threads,mysqladmin -uroot status|cut -f3 -d":"|cut -f2 -d" "
#UserParameter=mysql.questions,mysqladmin -uroot status|cut -f4 -d":"|cut -f2 -d" "
#UserParameter=mysql.slowqueries,mysqladmin -uroot status|cut -f5 -d":"|cut -f2 -d" "
#UserParameter=mysql.qps,mysqladmin -uroot status|cut -f9 -d":"|cut -f2 -d" "
#UserParameter=mysql.version,mysql -V
```

- *mysql.ping*

Check whether MySQL is alive.

```
    Result: 0 - not started 1 - alive
```

- *mysql.uptime*

Number of seconds MySQL is running.

- *mysql.threads*

Number of MySQL threads.

- *mysql.questions*

Number of processed queries.

- *mysql.slowqueries*

Number of slow queries.

- *mysql.qps*

Queries per second.

- *mysql.version*

Version of MySQL. For example: mysql Ver 14.14 Distrib 5.1.53, for pc-linux-gnu (i686)

For additional information see also the userparameter_mysql.conf file in conf/zabbix_agentd directory.

Mikrotik routers

Use SNMP agent provided by Mikrotik. See http://www.mikrotik.com [http://www.mikrotik.com] for more information.

WIN32

Use Zabbix W32 agent included (pre-compiled) into Zabbix distribution.

Novell

Use MRTG Extension Program for NetWare Server (MRTGEXT.NLM) agent for Novell. The agent is compatible with protocol used by Zabbix. It is available from http://forge.novell.com/modules/xfmod/project/?mrtgext [http://forge.novell.com/modules/xfmod/project/?mrtgext].

Items have to be configured of type Zabbix Agent and must have keys according to the MRTGEXT documentation.

For example:

- **UTIL1** - 1 minute average CPU utilization
- **CONNMAX** - max licensed connections used
- **VFKSys** - bytes free on volume Sys:

Full list of parameters supported by the agent can be found in readme.txt, which is part of the software.

Tuxedo

Tuxedo command line utilities tmadmin and qmadmin can be used in definition of a UserParameter in order to return per server/service/queue performance counters and availability of Tuxedo resources.

Informix

Standard Informix utility **onstat** can be used for monitoring of virtually every aspect of Informix database. Also, Zabbix can retrieve information provided by Informix SNMP agent.

HP OpenView

Zabbix can be configured to send messages to OpenView server. The following steps must be performed:

Step 1

Define new media.

The media will execute a script which will send required information to OpenView.

Step 2

Define new user.

The user has to be linked with the media.

Step 3

Configure actions.

Configure actions to send all (or selected) trigger status changes to the user.

Step 4

Write media script.

The script will have the following logic. If trigger is ON, then execute OpenView command *opcmsg -id application=<application> msg_grp=<msg_grp> object=<object> msg_text=<text>*. The command will return unique message ID which has to be stored somewhere, preferrably in a new table of ZABBIX database. If trigger is OFF then *opcmack <message id>* has to be executed with message ID retrieved from the database.

Refer to OpenView official documentation for more details about opcmsg and opcmack. The media script is not given here.

2.0/manual/appendix/recipes.txt · Last modified: 2012/09/04 14:42 by martins-v
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

10 Performance tuning

This is a work in progress.

Overview

It is very important to have Zabbix system properly tuned for maximum performance.

Hardware

General advice on hardware:

- Use fastest processor available
- SCSI or SAS is better than IDE (performance of IDE disks may be significantly improved by using utility hdparm) and SATA
- 15K RPM is better than 10K RPM which is better than 7200 RPM
- Use fast RAID storage
- Use fast Ethernet adapter
- Having more memory is always better

Operating system

- Use latest (stable!) version of OS
- Exclude unnecessary functionality from kernel
- Tune kernel parameters

Zabbix configuration parameters

Many parameters may be tuned to get optimal performance.

zabbix_server

StartPollers

General rule - keep value of this parameter as low as possible. Every additional instance of zabbix_server adds known overhead, in the same time, parallelism is increased. Optimal number of instances is achieved when queue, on average, contains minimum number of parameters (ideally, 0 at any given moment). This value can be monitored by using internal check zabbix[queue].

See the "See also" section at the bottom of this page to find out how to configure optimal count of zabbix processes.

DebugLevel

Optimal value is 3.

DBSocket

MySQL only. It is recommended to use DBSocket for connection to the database. That is the fastest and the

most secure way.

Database engine

This is probably the most important part of Zabbix tuning. Zabbix heavily depends on the availability and performance of database engine.

- use fastest database engine, i.e. MySQL
- use stable release of a database engine
- rebuild MySQL or PostgreSQL from sources to get maximum performance
- follow performance tuning instructions taken from MySQL or PostgreSQL documentation
- for MySQL, use InnoDB table structure
- ZABBIX works at least 1.5 times faster (comparing to MyISAM) if InnoDB is used. This is because of increased parallelism. However, InnoDB requires more CPU power.
- tuning the database server for the best performance is highly recommended.
- keep database tables on different hard disks
- 'history', 'history_str, 'items' 'functions', triggers', and 'trends' are most heavily used tables.
- for large installations, keeping of MySQL temporary files in tmpfs is recommended

General advice

- monitor required parameters only
- tune 'Update interval' for all items. Keeping a small update interval may be good for nice graphs, however, this may overload Zabbix
- tune parameters for default templates
- tune housekeeping parameters
- do not monitor parameters which return the same information.
- avoid the use of triggers with long period given as function argument. For example, max(3600) will be calculated significantly slower than max(60).

See also

1. How to configure optimal count of zabbix processes [http://blog.zabbix.com/monitoring-how-busy-zabbix-processes-are/457]

2.0/manual/appendix/performance_tuning.txt · Last modified: 2013/10/26 16:15 by zalex_ua
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

11 Version compatibility

Supported agents

Older agents from Zabbix 1.0, Zabbix 1.1.x, Zabbix 1.4.x, Zabbix 1.6.x and Zabbix 1.8.x can still be used with Zabbix 2.0. It does not require any configuration changes on agent side.

However, to take full advantage of new and improved items, improved performance and reduced memory usage, use the latest 2.0 agent.

Supported Zabbix proxies

Only Zabbix 2.0 proxies may be used with Zabbix server. Zabbix 1.6 and 1.8 proxies are not supported with Zabbix 2.0 server.

Zabbix 2.0 proxies may only be used with Zabbix 2.0 server. They will not work with 1.8 or older Zabbix server.

Supported distributed monitoring nodes

All distributed monitoring nodes must be of the same major version. 1.8 nodes are not supported together with 2.0 nodes.

Supported XML files

XML files, exported with 1.8, are supported for import in Zabbix 2.0.

2.0/manual/appendix/compatibility.txt · Last modified: 2012/11/14 10:00 by richlv
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

Zabbix API

The Zabbix API is a powerful tool for extending Zabbix, developing apps for your favorite platform or integrating it with third party software

This section provides an overview of the functions provided by the Zabbix API and will help you find your way around the available classes and methods.

Monitoring

The Zabbix API allows you to access history and other data gathered during monitoring.

History

Retrieve historical values gathered by Zabbix monitoring processes for presentation or further processing.

History API

Events

Retrieve events generated by triggers, network discovery and other Zabbix systems for more flexible situation management or third-party tool integration.

Event API

Service monitoring

Retrieve detailed service layer availability information about any IT service.

IT service SLA calculation

Configuration

The Zabbix API allows you to manage the configuration of your monitoring system.

Hosts and host groups

Manage host groups, hosts and everything related to them, including host interfaces, host macros and maintenance periods.

Host API | Host group API | Host interface API | User macro API | Maintenance API

Items and applications

Define items to monitor. Create or remove applications and assign items to them.

Item API | Application API

Triggers

Configure triggers to notify you about problems in your system. Manage trigger dependencies.

Trigger API

Graphs

Edit graphs or separate graph items for better presentation of the gathered data.

Graph API | Graph item API

Templates

Manage templates and link them to hosts or other templates.

Template API

Export and import

Export and import Zabbix configuration data for configuration backups, migration or large-scale configuration updates.

Configuration API

Low-level discovery

Configure low-level discovery rules as well as item, trigger and graph prototypes to monitor dynamic entities.

LLD rule API | Item prototype API | Trigger prototype API | Graph prototype API

Screens

Edit global and template-level screens or each screen item individually.

Screen API | Screen item API | Template screen API | Template screen item API

Actions and alerts

Define actions and operations to notify users about certain events or automatically execute remote commands. Gain access to information about generated alerts and their receivers.

Action API | Alert API

IT services

Manage IT services for service-level monitoring and retrieve detailed SLA information about any service.

IT service API

Maps

Configure maps to create detailed dynamic representations of your IT infrastructure.

Map API

Web monitoring

Configure web scenarios to monitor your web applications and services.

Web scenario API

Network discovery

Manage network-level discovery rules to automatically find and monitor new hosts. Gain full access to information about discovered services and hosts.

Discovery rule API | Discovery check API | Discovery host API | Discovery service API

Administration

With the Zabbix API you can change administration settings of your monitoring system.

Users

Add users that will have access to Zabbix, assign them to user groups and grant permissions. Configure media types and the ways users will receive alerts.

User API | User group API | Media type API | Media API

General

Change certain global configuration options.

Icon map API | Image API | User macro API

Proxies

Manage the proxies used in your distributed monitoring setup.

Proxy API

Scripts

Configure and execute scripts to help you with your daily tasks.

Script API

API information

Retrieve the version of the Zabbix API so that your application could use version-specific features.

API info API

2.0/manual/appendix/api/api.txt · Last modified: 2013/02/07 11:03 by Pavels

Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]

Internal audit log

quick draft

Zabbix frontend AND API (? → ZBX-4842 [https://support.zabbix.com/browse/ZBX-4842]) log all changes to the configuration.

(well, currently they don't log all actions…)

Zabbix server logs all changes to the configuration (like the ones performed by network discovery and active agent auto-registration). All changes by the server are logged as being performed by the guest user.

Audit log entries are kept forever - processes like housekeeper do not remove entries from it.

Internal audit log can be viewed at Administration -> Audit.

2.0/manual/appendix/audit.txt · Last modified: 2012/04/15 21:31 by zalex_ua
Except where otherwise noted, content on this wiki is licensed under the following license:CC Attribution-Noncommercial-Share Alike 3.0 Unported [http://creativecommons.org/licenses/by-nc-sa/3.0/]